An Eager Regression Method Based on Best Feature Projections

Tolga Aydın and H. Altay Güvenir

Department of Computer Engineering
Bilkent University
Ankara, 06533, TURKEY

Abstract. This paper describes a machine learning method, called *Regression by Selecting Best Feature Projections* (RSBFP). In the training phase, RSBFP projects the training data on each feature dimension and aims to find the predictive power of each feature attribute by constructing simple linear regression lines, one per each continuous feature and number of categories per each categorical feature. Because, although the predictive power of a continuous feature is constant, it varies for each distinct value of categorical features. Then the simple linear regression lines are sorted according to their predictive power. In the querying phase of learning, the best linear regression line and thus the best feature projection are selected to make predictions.

Keywords: Prediction, Feature Projection, Regression.

1 Introduction

Prediction has been one of the most common problems researched in data mining and machine learning. Predicting the values of categorical features is known as classification, whereas predicting the values of continuous features is known as regression. From this point of view, classification can be considered as a subcategory of regression. In machine learning, much research has been performed for classification. But, recently the focus of researchers has moved towards regression, since many of the real-life problems can be modeled as regression problems.

There are two different approaches for regression in machine learning community: Eager and lazy learning. Eager regression methods construct rigorous models by using the training data, and the prediction task is based on these models. The advantage of eager regression methods is not only the ability to obtain the interpretation of the underlying data, but also the reduced query time. On the other hand, the main disadvantage is their long train time requirement. Lazy regression methods, on the other hand, do not construct models by using the training data. Instead, they delay all processing to prediction phase. The most important disadvantage of lazy regression methods is the fact that, they do not provide an interpretable model of the training data, because the model is usually the training data itself. It is not a compact description of the training data, when compared to the models constructed by eager regression methods, such as regression trees and rule based regression.

In the literature, many eager and lazy regression methods exist. Among eager regression methods, CART [1], RETIS [7], M5 [5], DART [2], and Stacked

L. Monostori, J. Váncza, and M. Ali (Eds.): IEA/AIE 2001, LNAI 2070, pp. 217-226, 2001.

Regressions [9] induce regression trees, FORS [6] uses inductive logic programming for regression, RULE [3] induces regression rules, and MARS [8] constructs mathematical models. Among lazy regression methods, *k*NN [4, 10, 15] is the most popular nonparametric instance-based approach.

In this paper, we describe an eager learning method, namely *Regression by Selecting Best Feature Projections* (RSBFP) [13, 14]. This method makes use of the linear least squares regression.

A preprocessing phase is required to increase the predictive power of the method. According to the Chebyshev's result [12], for any positive number k, at least $(1 - 1/k^2) * 100\%$ of the values in any population of numbers are within k standard deviations of the mean. We find the standard deviation of the target values of the training data, and discard the training data whose target value is not within k standard deviations of the mean target. Empiricaly, we reach the best prediction by taking k as $\sqrt{2}$.

In the first phase, RSBFP constructs projections of the training data on each feature, and this phase continues by constructing simple linear regression lines, one per each continuous feature and number of categories per each categorical feature. Then, these simple linear regression lines are sorted according to their prediction ability. In the querying phase of learning, the target value of a query instance is predicted using the simple linear regression line having the minimum relative error, i.e. having the maximum predictive power. If this linear regression line is not suitable for our query instance, we keep searching for the best linear regression line among the ordered list of simple linear regression lines.

In this paper, RSBFP is compared with three eager (RULE, MARS, DART) and one lazy method (*k*NN) in terms of predictive power and computational complexity. RSBFP is better not only in terms of predictive power but also in terms of computational complexity, when compared to these well-known methods. For most data mining or knowledge discovery applications, where very large databases are in concern, this is thought of a solution because of low computational complexity. Again RSBFP is noted to be powerful in the presence missing feature values, target noise and irrelevant features.

In Section 2, we review the *k*NN, RULE, MARS and DART methods for regression. Section 3 gives a detailed description of the RSBFP. Section 4 is devoted to the empirical evaluation of RSBFP and its comparison with other methods. Finally, in Section 5, conclusions are presented.

2 Regression Overview

*k*NN is the most commonly used lazy method for both classification and regression problems. The underlying idea behind the *k*NN method is that the closest instances to the query point have similar target values to the query. Hence, the *k*NN method first finds the closest instances to the query point in the instance space according to a distance measure. Generally, the Euclidean distance metric is used to measure the similarity between two points in the instance space. Therefore, by using Euclidean distance metric as our distance measure, k closest instances to the query point are found. Then *k*NN outputs the distance-weighted average of the target values of those closest instances as the prediction for that query instance.

In machine learning, inducing rules from a given train data is also popular. Weiss and Indurkhya adapted the rule-based classification algorithm [11], Swap-1, for regression. Swap-1 learns decision rules in Disjunctive Normal Form (DNF). Since Swap-1 is designed for the prediction of categorical features, using a preprocessing procedure, the numeric feature in regression to be predicted is transformed to a nominal one. For this transformation, the P-class algorithm is used [3]. If we let $\{y\}$ be a set of output values, this transformation can be regarded as a one-dimensional clustering of training instances on response variable y, in order to form classes. The purpose is to make y values within one class similar, and across classes dissimilar. The assignment of these values to classes is done in such a way that the distance between each y_i and its class mean must be minimum. After formation of pseudo-classes and the application of Swap-1, a pruning and optimization procedure can be applied to construct an optimum set of regression rules.

MARS [8] method partitions the training set into regions by splitting the features recursively into two regions, by constructing a binary regression tree. MARS is continuous at the borders of the partitioned regions. It is an eager, partitioning, interpretable and an adaptive method.

DART, also an eager method, is the latest regression tree induction program developed by Friedman [2]. It avoids limitations of disjoint partitioning, used for other tree-based regression methods, by constructing overlapping regions with increased training cost.

3 Regression by Selecting Best Feature Projections (RSBFP)

RSBFP method tries to determine the feature projection that achieves the highest prediction accuracy. The next subsection describes the training phase for RSBFP, then we describe the querying phase.

3.1 Training

Training in RSBFP begins simply by storing the training data set as projections to each feature separately. A copy of the target values is associated with each projection and the training data set is sorted for each feature dimension according to their feature values. If a training instance includes missing values, it is not simply ignored as in many regression algorithms. Instead, that training instance is stored for the features on which its value is given. The next step involves constructing the simple linear regression lines for each feature. This step differs for categorical and continuous features. In the case of continuous features, exactly one simple linear regression line per feature is constructed. On the other hand, the number of simple linear regression lines per each categorical feature is the number of distinct feature values at the feature of concern. For any categorical feature, the parametric form of any simple regression line is constant, and it is equal to the average target value of the training instances whose corresponding feature value is equal to that categorical value. The training phase continues by sorting these regression lines according to their predictive power. The training phase can be illustrated through an example.

Let our example domain consist of four features, f_1, f_2, f_3 and f_4, where f_1, f_2 are continuous and f_3, f_4 are categorical. For continuous features, we define *minvalue*[*f*] and *maxvalue*[*f*] to denote the minimum and maximum value of feature *f*, respectively. For categorical features, *No_categories* [*f*] is defined to give the number of distinct categories of feature *f*. In our example domain, let the following values be observed:

minvalue[f_1] = 4, *maxvalue*[f_1] = 10
minvalue[f_2] = 2, *maxvalue*[f_2] = 8
No_categories [f_3] = 2 (values: A, B)
No_categories [f_4] = 3 (values: X, Y, Z)

For this example domain, 7 simple linear regression lines are constructed: 1 for f_1, 1 for f_2, 2 for f_3, and finally 3 for f_4. Let the following be the parametric form of the simple linear regression lines:

Simple linear regression line for f_1: target = $2f_1$ - 5
Simple linear regression line for f_2: target = $-4f_2$ + 7
Simple linear regression line for A category of f_3: target = 6
Simple linear regression line for B category of f_3: target = -5
Simple linear regression line for X category of f_4: target = 10
Simple linear regression line for Y category of f_4: target = 1
Simple linear regression line for Z category of f_4: target = 12

The training phase is completed by sorting these simple linear regression lines according to their predictive accuracy. The relative error (RE) of the regression lines is used as the indicator of predictive power: the smaller the RE, the stronger the predictive power. The RE of a simple linear regression line is computed by the following formula:

$$RE = \frac{MAD}{\frac{1}{Q}\sum_{i=1}^{Q}|t(q_i) - \bar{t}|}$$

where Q is the number of training instances used to construct the simple linear regression line, $\bar{t}$ is the median of the target values of Q training instances, $t(q_i)$ is the actual target value of the i^{th} training instance. The MAD (Mean Absolute Distance) is defined as follows:

$$MAD = \frac{1}{Q}\sum_{i=1}^{Q}|t(q_i) - \hat{t}(q_i)|$$

Here, $\hat{t}$ (q_i) denotes the predicted target value of the i^{th} training instance according to the induced simple linear regression line.

We had 7 simple linear regression lines, and let's suppose that they are sorted in the following order, from the best predictive to the worst one:

f_3=A > f_4=X > f_2 > f_1 > f_4=Y > f_4=Z > f_3=B

This shows that any categorical feature's predictive power may vary among its categories. For the above sorting schema, categorical feature f_3 's predictions are reliable among its category A, although it is very poor among category B.

3.2 Querying

In order to predict the target value of a query instance t_i, the RSBFP method uses exactly one linear regression line. This line may not always be the best one. The reason for this situation is explained via an example. Let the feature values of the query instance t_i be as the following:

$$f_1(t_i) = 5, \quad f_2(t_i) = 10, \quad f_3(t_i) = \text{B}, \quad f_4(t_i) = \text{missing}$$

Although the best linear regression line is f_3=A, this line can not be used for our t_i , since $f_3(t_i) \neq$ A. The next best linear regression line, which is worse than only f_3=A, is f_4=X. This line is also inappropriate for our t_i. No prediction can be made for missing feature values ($f_4(t_i)$ = missing). Therefore, the search for the best linear regression line continues. The line constructed by f_2 comes next. It is again not possible to benefit from this simple linear regression line. Because $f_2(t_i) = 10$, and it is not in the range of f_2, (2,8). Fortunately, we find an appropriate regression line in the fourth trial. Our $f_1(t_i)$, which is 5, is in the range of f_1, (4,10). So the prediction made for target value of t_i is $(2 * f_1(t_i) - 5) = (2 * 5 - 5) = 5$. Once the appropriate linear regression line is found, remaining linear regression lines need not be dealed anymore.

4 Empirical Evaluation

RSBFP method was compared with the other well-known methods mentioned above, in terms of predictive accuracy and time complexity. We have used a repository consisting of 26 data files in our experiments. The characteristics of the data files are summarized in Table1. Most of these data files are used for the experimental analysis of function approximation techniques and for training and demonstration by machine learning and statistics community.

10 fold cross-validation technique was employed in the experiments. For lazy regression method k parameter was taken as 10, where k denotes the number of nearest neighbors considered around the query instance.

In terms of predictive accuracy, RSBFP performed the best on 9 data files among the 26, and obtained the lowest mean relative error (Table 2).

In terms of time complexity, RSBFP performed the best in the total (training + querying) execution time, and became the fastest method (Table 3, 4).

In machine learning, it is very important for an algorithm to still perform well when noise, missing feature value and irrelevant features are added to the system. Experimental results showed that RSBFP was again the best method whenever we added 20% target noise, 20% missing feature value and 30 irrelevant features to the

system, by having the lowest mean relative errors. RSBFP performed the best on 7 data files in the presence of 20% missing feature value, the best on 21 data files in the presence of 20% target noise and the best on 10 data files in the presence of 30 irrelevant features (Table 5, 6, 7).

5 Conclusions

In this paper, we have presented an eager regression method based on selecting best feature projections. RSBFP is better than other well-known eager and lazy regression methods in terms of prediction accuracy and computational complexity. It also enables the interpretation of the training data. That is, the method clearly states the best feature projections that are powerful enough to determine the value of the target feature.

The robustness of any regression method can be determined by analyzing the predictive power of that method in the presence of target noise, irrelevant features and missing feature values. These three factors heavily exist in real life databases, and it is important for a learning algorithm to give promising results in the presence of those factors. Empirical results indicate that RSBFP is also a robust method.

Table 1. Characteristics of the data files used in the empirical evaluations, C: Continuous, N: Nominal

Dataset	Original Name	Instances	Features (C+N)	Missing Values
AB	Abalone	4177	8 (7 + 1)	None
AP	Airport	135	4 (4 + 0)	None
AU	Auto-mpg	398	7 (6 + 1)	6
BA	Baseball	337	16 (16 + 0)	None
BU	Buying	100	39 (39 + 0)	27
CH	Cpu	209	7 (6 + 1)	None
CN	Country	122	20 (20 + 0)	34
ED	Education	1500	43 (43 + 0)	2918
EL	Electric	240	12 (10 + 2)	58
FA	Fat	252	17 (17 + 0)	None
FC	Fishcatch	158	7 (6 + 1)	87
FF	Fruitfly	125	4 (3 + 1)	None
HO	Housing	506	13 (12 + 1)	None
HR	Home Run Race	163	19 (19 + 0)	None
NE	Northridge	2929	10 (10 + 0)	None
NT	Normal Temp.	130	2 (2 + 0)	None
PL	Plastic	1650	2 (2 + 0)	None
PV	Poverty	97	6 (5 + 1)	6
RE	Read	681	25 (24 + 1)	1097
S2	Solar Flare	1066	10 (0 + 10)	None
SC	Schools	62	19 (19 + 0)	1
SE	Servo	167	4 (0 + 4)	None
SP	Stock Prices	950	9 (9 + 0)	None
TV	Televisions	40	4 (4 + 0)	None
US	Usnews Coll.	1269	31 (31 + 0)	7624
VL	Villages	766	32 (29 + 3)	3986

Table 2. Relative errors (REs) of algorithms. Best REs are shown in bold font

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	0.729	**0.661**	0.899	0.683	0.678
AP	0.550	0.612	0.744	0.720	**0.546**
AU	0.489	**0.321**	0.451	0.333	0.346
BA	0.768	**0.443**	0.666	0.493	0.508
BU	**0.678**	0.961	0.946	0.947	0.896
CH	0.781	0.944	0.678	0.735	**0.510**
CN	**1.429**	1.642	6.307	5.110	1.695
ED	0.668	0.654	**0.218**	0.359	0.410
EL	**1.003**	1.194	1.528	1.066	1.118
FA	0.725	0.785	0.820	**0.305**	0.638
FC	0.578	0.697	0.355	**0.214**	0.415
FF	1.016	1.201	1.558	**1.012**	1.077
HO	0.698	0.600	0.641	0.526	**0.522**
HR	0.890	0.907	0.890	**0.769**	0.986
NE	0.969	1.034	1.217	0.928	**0.873**
NT	**0.976**	1.232	1.250	1.012	1.112
PL	0.887	0.475	0.477	**0.404**	0.432
PV	0.921	0.796	0.916	1.251	**0.691**
RE	**0.997**	1.062	1.352	1.045	1.189
S2	**1.434**	2.307	1.792	1.556	1.695
SC	0.376	0.388	0.341	**0.223**	0.352
SE	0.868	0.619	**0.229**	0.432	0.337
SP	1.416	**0.599**	0.906	0.781	0.754
TV	**1.176**	1.895	4.195	7.203	2.690
US	**0.402**	0.480	0.550	0.412	0.623
VL	**0.940**	1.017	1.267	1.138	1.355
Mean	**0.860**	0.905	1.200	1.140	0.864

Table 3. Train time of algorithms in milliseconds. Best results are shown in bold font

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	148	**8.9**	3219	10270	477775
AP	1.1	**0**	90.8	159.2	62
AU	8.9	**0.6**	248.9	570.5	1890.1
BA	19	**0**	181.8	915.1	3171.1
BU	10.5	**0**	67.1	761.7	794.4
CH	4.1	**0**	52.7	575.3	286
CN	8	**0.1**	108.6	475.3	481
ED	278.2	**13.5**	862.8	10143.9	27266
EL	8.1	**0.2**	69.5	407.5	1017
FA	15.8	**0**	161.1	985	1773.9
FC	2.1	**0**	47.8	240.2	201.4
FF	1.1	**0**	34.1	99.5	45.9
HO	21.2	**1**	264.9	1413.9	8119.7
HR	8.2	**0**	57.5	616.3	893.9
NE	130.5	**7.4**	3493	5709.9	87815
NT	**0**	**0**	30.6	69.3	18.9
PL	10	**0.2**	175.3	824.8	10024.4
PV	1	**0**	40.9	127.3	44
RE	52	**3**	196	2744.6	33044.6
S2	36.1	**3.5**	108.8	667.2	971.4
SC	3	**0**	45.3	260.8	84.4
SE	1.8	**0**	37	116.4	83.4
SP	28.5	**1.4**	365.1	2281.4	17346.4
TV	**0**	**0**	30.9	31.1	3.1
US	136.4	**7.4**	2547.1	8435.2	168169
VL	74.6	**4.4**	513.6	3597.8	23405
Mean	38.777	**1.985**	501.93	2019.2	33261

Table 4. Query time of algorithms in milliseconds. Best results are shown in bold font

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	23.3	6547	14433.1	7.9	**6.1**
AP	1	3.4	141.7	**0**	**0**
AU	3	64.5	462.2	**0**	**0**
BA	2.1	54.6	244.8	**0**	**0**
BU	**0**	11.6	32.1	**0**	**0**
CH	1	11.6	87.3	**0**	**0**
CN	1	8.4	98.4	**0**	0.1
ED	9	2699.7	312.3	2.7	**1.7**
EL	1.6	21	117.5	**0**	**0**
FA	2	33.1	96.4	**0**	**0**
FC	1.1	7.9	48.8	**0**	**0**
FF	0.9	2	45.4	**0**	**0**
HO	3.2	107.8	410.5	**0**	**0**
HR	0.6	13.3	43	**0**	**0**
NE	14	3399.4	11326.8	4.7	**1.75**
NT	1	1.9	30.8	**0**	**0**
PL	16.7	571.9	2192.7	**0.2**	1.2
PV	**0**	2.2	37.1	**0**	**0**
RE	3	265.6	627.2	**0**	1
S2	4	407.8	223.6	0.4	**0**
SC	**0**	2	27.8	3.7	**0**
SE	0.1	4.2	49.1	**0**	**0**
SP	6.2	303.2	1090.9	0.1	**0**
TV	**0**	**0**	24	**0**	**0**
US	8	1383.2	1877.3	7	**2**
VL	6	439	1118.2	0.3	**0**
Mean	4.184	629.47	1353.8	1.038	**0.533**

Table 5. REs of algorithms, where 20% missing feature value are added. Best Res are shown in bold font (* Means result isn't available due to singular variance/covariance matrix)

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	0.729	0.750	0.961	0.748	**0.688**
AP	0.562	0.726	0.676	0.798	**0.546**
AU	0.500	0.414	0.526	0.414	**0.363**
BA	0.785	**0.553**	0.833	0.637	0.576
BU	**0.785**	0.951	0.878	0.862	1.026
CH	0.746	0.922	0.832	0.747	**0.608**
CN	**1.480**	1.856	3.698	3.733	2.377
ED	0.685	0.743	**0.497**	0.595	0.536
EL	**1.005**	1.097	1.537	1.073	1.191
FA	0.749	0.849	0.948	**0.731**	0.735
FC	0.570	0.675	0.543	0.537	**0.401**
FF	1.019	1.711	1.557	**1.012**	1.347
HO	0.718	0.761	0.748	0.649	**0.590**
HR	0.899	0.910	1.040	**0.836**	0.974
NE	0.974	1.072	1.272	**0.972**	*
NT	1.020	1.229	1.363	**0.989**	1.222
PL	0.903	0.733	0.686	0.679	**0.420**
PV	0.920	0.976	1.189	1.026	**0.792**
RE	**0.996**	1.059	1.364	1.048	1.229
S2	1.429	1.851	1.751	1.557	**1.421**
SC	0.409	0.449	0.500	**0.303**	0.370
SE	0.879	0.921	0.849	0.746	**0.495**
SP	1.430	0.744	0.904	0.930	**0.707**
TV	**1.272**	4.398	3.645	16.50	2.512
US	**0.460**	0.558	0.620	0.497	0.844
VL	**0.949**	1.056	1.410	1.090	*
Mean	**0.880**	1.086	1.186	1.527	0.920

Table 6. REs of algorithms, where 20% target noise are added. Best REs are shown in bold font

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	**0.819**	7.592	9.301	7.602	6.603
AP	0.952	0.807	1.122	0.856	**0.785**
AU	**0.488**	1.832	2.531	2.107	1.981
BA	0.813	**0.457**	0.712	0.537	0.556
BU	**0.597**	12.66	12.92	13.30	10.67
CH	0.815	0.930	0.782	0.745	**0.636**
CN	**1.516**	1.676	3.102	5.874	2.040
ED	**0.653**	2.166	2.384	2.164	2.276
EL	**0.978**	1.465	1.899	1.148	1.431
FA	**0.684**	2.525	3.208	2.447	2.058
FC	0.544	0.710	0.528	0.501	**0.387**
FF	**1.030**	2.394	3.247	1.710	2.089
HO	**0.865**	2.801	3.635	2.893	2.611
HR	**0.863**	7.853	11.53	10.29	6.115
NE	**0.986**	38.84	42.32	37.66	31.54
NT	**0.951**	1.403	2.220	1.037	1.196
PL	**0.852**	5.492	5.777	4.921	5.107
PV	**0.829**	9.429	9.456	4.213	6.038
RE	**0.952**	6.597	10.33	6.759	7.108
S2	**2.366**	73.89	77.21	70.90	71.40
SC	**0.538**	0.583	0.968	0.700	0.627
SE	**0.697**	21.29	27.77	22.01	21.72
SP	**1.183**	1.921	3.887	1.966	1.871
TV	**1.468**	2.087	4.569	7.267	2.671
US	0.643	0.636	0.865	**0.541**	0.764
VL	**0.973**	1.030	1.513	0.977	1.518
Mean	**0.925**	8.041	9.377	8.120	7.378

Table 7. REs of algorithms, where 30 irrelevant features are added. Best Res are shown in bold font (* Means result isn't available due to singular variance/covariance matrix)

Dataset	RSBFP	KNN	RULE	MARS	DART
AB	0.728	0.873	0.934	**0.682**	*
AP	**0.555**	1.514	0.723	0.682	0.657
AU	0.488	0.538	0.491	**0.368**	0.511
BA	0.768	0.568	0.574	**0.536**	0.628
BU	**0.678**	0.968	1.073	0.877	0.969
CH	0.781	1.107	0.753	**0.613**	0.668
CN	**1.425**	2.854	1.794	4.126	1.662
ED	0.668	0.802	**0.268**	0.404	0.573
EL	**1.006**	1.037	1.367	1.134	1.236
FA	0.725	1.026	1.039	**0.249**	0.877
FC	0.578	0.917	0.456	**0.247**	0.420
FF	**1.030**	1.063	1.513	1.777	1.430
HO	0.698	0.920	0.701	**0.521**	0.653
HR	0.890	0.932	1.049	**0.847**	1.165
NE	0.969	1.076	1.284	**0.916**	*
NT	**1.000**	1.079	1.484	1.370	1.156
PL	0.887	0.961	0.575	**0.407**	0.734
PV	0.966	**0.855**	0.934	1.005	1.013
RE	**0.998**	1.045	1.380	1.042	1.311
S2	**1.433**	1.454	1.765	1.629	1.490
SC	0.376	0.582	0.386	**0.305**	0.391
SE	0.926	0.835	**0.471**	0.798	0.641
SP	1.416	1.188	0.914	0.817	**0.756**
TV	**1.220**	3.241	5.572	5.614	2.709
US	0.402	0.757	0.557	**0.394**	0.906
VL	**0.939**	1.050	1.454	1.257	1.307
Mean	**0.867**	1.124	1.135	1.100	1.000

References

[1] Breiman, L, Friedman, J H, Olshen, R A and Stone, C J '*Classification and Regression Trees'* Wadsworth, Belmont, California (1984)
[2] Friedman, J H 'Local Learning Based on Recursive Covering' Department of Statistics, Stanford University (1996)
[3] Weiss, S and Indurkhya, N ' Rule-based Machine Learning Methods for Functional Prediction' *Journal of Artificial Intelligence Research* Vol 3 (1995) pp 383-403
[4] Aha, D, Kibler, D and Albert, M 'Instance-based Learning Algorithms' *Machine Learning* Vol 6 (1991) pp 37 – 66
[5] Quinlan, J R 'Learning with Continuous Classes' *Proceedings AI'92* Adams and Sterling (Eds) Singapore (1992) pp 343-348
[6] Bratko, I and Karalic A 'First Order Regression' *Machine Learning* Vol 26 (1997) pp 147-176
[7] Karalic, A 'Employing Linear Regression in Regression Tree Leaves' *Proceedings of ECAI'92* Vienna, Austria, Bernd Newmann (Ed.) (1992) pp 440-441
[8] Friedman, J H 'Multivariate Adaptive Regression Splines' *The Annals of Statistics* Vol 19 No 1 (1991) pp 1-141
[9] Breiman, L 'Stacked Regressions' *Machine Learning* Vol 24 (1996) pp 49-64
[10] Kibler, D, Aha D W and Albert, M K 'Instance-based Prediction of Real-valued Attributes' *Comput. Intell*. Vol 5 (1989) pp 51-57
[11] Weiss, S and Indurkhya, N 'Optimized Rule Induction' *IEEE Expert* Vol 8 No 6 (1993) pp 61-69
[12] Graybill, F, Iyer, H and Burdick, R 'Applied Statistics' Upper Saddle River, NJ (1998)
[13] Aydın, T 'Regression by Selecting Best Feature(s)' *M.S.Thesis*, Computer Engineering, Bilkent University, September, (2000)
[14] Aydın, T and Güvenir, H A 'Regression by Selecting Appropriate Features' *Proceedings of TAINN'2000*, Izmir, June 21-23, (2000), pp 73-82
[15] Uysal, İ and Güvenir, H A 'Regression on Feature Projections' *Knowledge-Based Systems*, Vol.13, No:4, (2000), pp 207-214